

EMG 데이터를 이용한 머신러닝 기반 실시간 제스처 분류 연구[☆]

A Study on Machine Learning-Based Real-Time Gesture Classification Using EMG Data

박 하 제¹ 양 희 영¹ 최 소 진¹ 김 대 연² 남 춘 성^{1*}
Ha-Je Park Hee-Young Yang So-Jin Choi Dae-Yeon Kim Choon-Sung Nam

요 약

사용자가 제스처를 통해 입력을 할 수 있는 방안들 중에서 근전도(EMG, Electromyography)를 통한 제스처 인식은 근육 내 작은 전극을 통해 사용자의 움직임에 감지하고 이를 입력 방법으로 사용할 수 있는 방법이다. EMG 데이터를 통해 사용자 제스처를 분류하기 위해서는 사용자로부터 수집된 EMG Raw 데이터를 머신러닝으로 학습하여야 하는데 이를 위해서는 EMG 데이터를 전처리 과정을 통해 특징을 추출하여야 한다. EMG 특성은 IEMG(Integrated EMG), MAV(Mean Absolute Value), SSI(Simple Square Integral), VAR(VARiance), RMS(Root Mean Square) 등과 같은 수식을 통해서 나타낼 수 있다. 또한, 제스처를 입력으로 사용하기 위해서는 사용자가 입력하는 데 필요한 지각, 인지, 반응에 필요한 시간을 기준으로 제스처 분류가 가능한 시간을 알아내야 한다. 이를 위해 최대 1,000ms에서 최소 100ms까지 세그먼트 사이즈를 변화시켜 특징을 추출 후 제스처 분류가 가능한 세그먼트 사이즈를 찾아낸다. 특히 데이터 학습은 overlapped segmentation 방법을 통해 데이터와 데이터 사이 간격을 줄여 학습 데이터 개수를 늘린다. 이를 통해 KNN, SVC, RF, XGBoost 4가지 머신러닝 방식을 통해 이를 학습하고 결과를 도출한다. 실험 결과 실시간으로 사용자의 제스처 입력이 가능한 최대 세그먼트 사이즈인 200ms에서 KNN, SVC, RF, XGboost 4가지 모든 모델에서 96% 이상의 정확도를 도출하였다.

☞ 주제어 : EMG, 머신러닝, 실시간 제스처 분류, EMG 데이터 전처리

ABSTRACT

This paper explores the potential of electromyography (EMG) as a means of gesture recognition for user input in gesture-based interaction. EMG utilizes small electrodes within muscles to detect and interpret user movements, presenting a viable input method. To classify user gestures based on EMG data, machine learning techniques are employed, necessitating the preprocessing of raw EMG data to extract relevant features. EMG characteristics can be expressed through formulas such as Integrated EMG (IEMG), Mean Absolute Value (MAV), Simple Square Integral (SSI), Variance (VAR), and Root Mean Square (RMS). Additionally, determining the suitable time for gesture classification is crucial, considering the perceptual, cognitive, and response times required for user input. To address this, segment sizes ranging from a minimum of 100ms to a maximum of 1,000ms are varied, and feature extraction is performed to identify the optimal segment size for gesture classification. Notably, data learning employs overlapped segmentation to reduce the interval between data points, thereby increasing the quantity of training data. Using this approach, the paper employs four machine learning models (KNN, SVC, RF, XGBoost) to train and evaluate the system, achieving accuracy rates exceeding 96% for all models in real-time gesture input scenarios with a maximum segment size of 200ms.

☞ keyword : EMG, Machine learning, Real time gestuer classification, EMG data preprocessing

1. 서 론

다양한 IT 기기의 출현과 함께 다양한 방식의 인터페이스(Interface)가 사용되고 있다. 대표적인 인터페이스로 키보드, 마우스, 터치, 음성, 제스처 등과 같은 다양한 인터페이스가 존재한다[1]. 이 중에서도 제스처를 통한 입력 방법은 다른 인터페이스보다 사용자 중심적인 인터페이스로 더욱 직관적인 방법이다[2]. 또한, 제스처 입력 방식은 Apple 사의 Vision Pro[3]와 같은 AR/VR 환경에서 별도의 컨트롤러(Controller)를 필요하지 않고, 공간적인

¹ Department of Software Convergence Engineering, Inha University, Incheon, 15798, Korea

² Department of Mechatronics, Inha University, Incheon, 15798, Korea

* Corresponding author (namgun99@inha.ac.kr)

[Received 23 November 2023, Reviewed 2 January 2024(R2 11 March 2024), Accepted 21 March 2024]

☆ 본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성사업의 연구결과로 수행되었음 (IITP-2023-RS-2023-00259678)

제약 상황에서 적합한 인터페이스 방식이다.

제스처를 인식하기 위한 방법으로는 근전도(EMG, Electromyography) 데이터를 이용한 방법이 있다[4]. EMG는 근육 내의 작은 전극을 통해 측정되며 이 전극은 근육 섬유 주변에 배치되어 전기적 활동을 감지하고 측정한다. 근육 내에서 신경 자극이 전달되면 근육 섬유는 수축하고, 이 과정에서 전기적인 활동이 발생한다. 특정 근육의 활동 패턴이나 근육군의 협동 작용에 따라 EMG의 패턴이 달라지는데, 이러한 특성을 활용하여 제스처마다 특정 근육이 수축하거나 이완하는 패턴을 분석하고 학습함으로써 제스처를 분류할 수 있다.

EMG는 의료, 제어, 시각 언어 분야와 같이 다양한 분야에서 연구되고 있다[5]. 의료 분야에서 EMG는 인간의 행동 분석 및 행동 모방을 위해 연구되고 있다. 또한, 상지 절단 환자들의 의수를 제어하기 위한 방법으로 EMG 특징 추출과 데이터 추출 시간 사이의 관계에 대한 연구도 진행되고 있다[6]. 제어 분야에서는 의수 및 의족뿐만 아니라 전자기기의 인터페이스를 제어하기 위하여 연구되고 있다. 그리고 시각 언어 분야는 의사소통을 위한 손동작이나 수화 등을 인식하는 기술에 대해 연구되고 있다[7,8]. 이러한 연구들에서 EMG 데이터를 통해 특징을 분류하기 위해 가장 중요한 것은 수집시간에 따른 데이터의 양이다. 또한, EMG 데이터 처리 방법에 따라 분류 결과가 달라질 수 있다. EMG 데이터의 처리 방법은 특징 추출 방법과 특징 분류 방법이 가장 많이 연구되고 있다. 특징 추출 방법에서는 EMG Raw 데이터를 다양한 수식으로 변환하여 EMG의 특징을 명확하게 나타내 분류 정확도를 향상할 수 있다[9,10]. 특징 분류 방법에서는 머신러닝으로 특징을 학습하여 분류가 가능하다[11]. 머신러닝을 통해서 학습할 수 있는 EMG 데이터는 수집 시간이 길수록 데이터의 분류 정확도는 높아지지만, 데이터의 양이 늘어나 분류 속도가 느려지기 때문에 실시간 처리가 어렵다. 반대로, 수집 시간이 짧으면 데이터의 양이 줄어들어 정확도가 낮아지지만, 분류의 속도가 빨라지기 때문에 실시간 처리에 유리하다[12]. 즉, 제스처를 실시간으로 입력하기 위해서는 데이터의 수집 시간을 줄이면서 제스처 분류의 정확도가 높게 유지할 수 있는 방안이 필요하다.

따라서 본 논문에서는 사용자 입력이 가능한 시간을 알아내기 위해 1,000ms에서 100ms까지 사용자 데이터 입력 시간에 따른 정확도를 비교 평가한다. 이를 통해 사용자가 실시간으로 제스처 입력이 가능한 시간을 제시한다. 또한, 머신러닝을 학습하기 위해 수집된 데이터에서

더 많은 양의 데이터를 추출하여 정확도를 높이기 위해 overlapped segmentation 방법을 적용한다.

2. 관련연구

2.1 데이터 수집을 위한 장치 : Myo Armband

본 논문에서는 사용자의 6가지 제스처의 데이터를 수집하기 위해 그림 1에 Thalmic Lab社의 Myo Armband를 사용하였다[13]. 이 장비의 샘플링 레이트는 200Hz이므로 초당 200개의 샘플(sample) 데이터를 수집하며, 총 8개의 센서로 초당 1,600개의 샘플 데이터를 수집한다. Myo Armband는 자체적으로 EMG 센서를 통해 측정된 데이터를 잡음 제거(noise filtering)하여 변환하여 출력한다.



(그림 1) Myo Armband
(Figure 1) Myo Armband

2.2 데이터 전처리 방안

시계열 데이터에 특징을 가진 EMG 데이터는 시계열 데이터 처리 방법을 이용하여 전처리하여 특징을 추출한다. 또한, EMG 특징 추출 방법에 따라 분류 정확도 차이가 있다. 따라서 많이 사용되는 15가지 전처리 수식인 Integrated EMG (IEMG), Mean Absolute Value (MAV), Modified Mean Absolute Value 1 (MAV1), Modified Mean Absolute Value 2 (MAV2), Mean Absolute Value Slope (MAVS), Simple Square Integral (SSI), Variance (VAR), Root Mean Square (RMS), Waveform length (WL), Zero crossing (ZC), Slope Sign Change (SSC), Willison amplitude (WAMP), Auto-regressive (AR) coefficients, Median Frequency (MDF), Mean Frequency (MNF)를 이용한 EMG 데이터 특징을 추출한다. 이와 같은 수식은 기존 연구 [9,10]에서 제시한 방안을 사용한다.

예를 들어 IEMG는 절대값의 합을 나타내는 지표로 수식 (1)과 같이 계산될 수 있다.

$$IEMG = \sum_{n=1}^N |x_n| \quad (1)$$

N 은 EMG의 전체 샘플 수를 나타내며, x_n 은 각 시간 단위의 EMG 샘플을 나타낸다. 각 시간 단위의 EMG 샘플의 절대값을 취하고 이를 모두 합산하여 EMG의 총 활동량을 계산한다.

수식 (2)는 MAV로 EMG의 평균 절대값을 나타낸다.

$$MAV = \frac{1}{N} \sum_{n=1}^N |x_n| \quad (2)$$

N 은 EMG의 전체 샘플 수를 나타내며, x_n 은 각 시간 단위의 EMG 샘플을 나타낸다. 각 시간 단위의 EMG 샘플의 절대값을 취하고 이를 모두 합산하여 전체 샘플수로 나눠 EMG의 평균 활동량을 계산한다.

2.3 머신러닝 학습 방안

특징을 추출한 EMG 데이터의 전처리 방법별 분류 정확도를 확인하기 위해 K-Nearest Neighbors(KNN), C-Support Vector Classification(SVC), Random Forest(RF), Extreme Gradient Boost(XGBoost) 총 4가지 머신러닝을 사용한 성능을 비교하고 EMG 데이터의 제스처 분류에 가장 적합한 머신러닝 알고리즘을 도출한다.

2.3.1 KNN(K-Nearest Neighbors)

KNN은 비모수 적 방식의 머신러닝 알고리즘으로, 해당 데이터와 가장 가까운 이웃 데이터들을 찾아 분류를 수행한다. 비슷한 특성이나 속성을 가진 데이터의 군집성을 고려하여 분류를 수행하기 때문에 센서별 EMG 데이터의 특징점을 이용하여 제스처 식별이 유용하다[14].

2.3.2 SVC(C-Support Vector Classification)

서포트 벡터 머신(Support Vector Machine, SVM)의 변형인 SVC는 여러 커널에 기반을 두고 있으며, 분류 문제를 해결하기 위한 단순하고 빠른 분류 모델들을 사용하면서도 안정적으로 결정 경계를 찾아 분류를 수행하는 머신러닝 알고리즘이다. 제스처별 EMG 데이터의 특징을

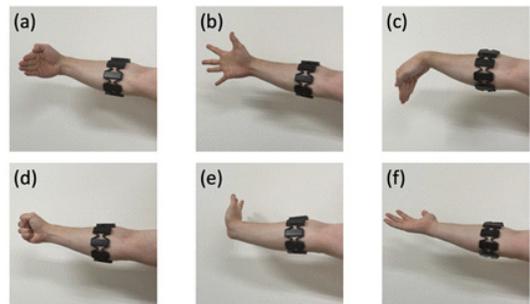
구분하는 것은 분류의 문제이기 때문에 SVC는 제스처를 분류하기 위한 최적의 결정 경계를 찾아 제스처 간의 차이를 명확하게 인식하고 분류할 수 있다[15].

2.3.3 RF(Random Forest)

RF는 앙상블 기법을 사용하여 결정 트리의 특정 특성에 따라 데이터를 분류해 한 번에 하나의 변수만을 고려하여 변수 간 상호작용을 파악하기가 어렵고 과적합이 생기는 단점을 개선해 성능이 향상된 머신러닝 알고리즘이다. RF는 특정 특성을 선택하는 트리를 여러 개 생성하여 이들을 기반으로 작업을 수행한다. 그리고 생성된 수많은 의사결정나무의 결과를 바탕으로 최종 답을 결정하여 결측치나 이상치에 강하다는 장점이 있으며, 과적합에도 강하다는 특징이 있다. 그렇기 때문에 이상치가 존재하는 시계열 데이터인 EMG 데이터를 안정적으로 제스처 분류를 수행하는 데에 유용하다[16].

2.3.4 XGBoost(Extreme Gradient Boosting)

XGBoost는 경사 부스팅(Gradient Boosting) 알고리즘을 기반으로 한 앙상블 학습 기법의 하나이다. XGBoost는 경사 부스팅의 단점인 느린 학습 시간과 과적합 문제를 보완한 알고리즘이다. 높은 성능과 자원 효율성을 보이며 병렬 처리 학습을 지원하여 처리 속도가 빠르다는 이점이 있다[16].



(그림 2) 6가지 손 제스처 (a) 손을 안으로 꺾는 동작, (b) 손을 펴는 동작, (c) 휴식 상태, (d) 주먹을 쥐는 동작, (e) 손을 밖으로 꺾는 동작, (f) 손바닥을 위로 향하는 동작

(Figure 2) 6 hand gesture : (a) wrist flexion (b) finger extension, (c) hand at rest, (d) hand close, (e) wrist extension, (f) forearm supination

3. 실험방법

3.1 제스처 EMG 데이터 수집 방법

측정에 참여한 연구대상자는 20대 연령을 가진 5명 (남:3명, 여:2명)의 지원자를 대상으로 하여 성별에 다양성을 주고 EMG 측정에 불편함이 없는 대상으로 선정하였다. Myo Armband 착용 방법은 Myo Armband의 전원 표시가 팔오금 중앙에 위치하도록 착용하였다. Myo Armband는 내부의 8개 EMG 센서로 측정된 EMG 데이터를 블루투스로 전송한다[12]. Thalmic Lab社에서 제공하는 SDK를 활용하여 C++ 프로그램 개발하여 EMG를 측정하였다. 측정에 사용된 동작은 사람의 의도 반영, 손짓의 용이성, 근육의 움직임에 고려하여 제스처를 정의하였다. 그림 2와 같이 손을 안으로 쥐는 동작(a), 손을 펴는 동작(b), 휴식 상태(c), 주먹을 쥐는 동작(d), 손을 밖으로 쥐는 동작(e), 손바닥을 위로 향하는 동작(f)으로 총 6가지의 동작이다. 각 동작 당 2초간 12번씩 데이터를 측정하였으며 총 데이터의 개수는 360개 (5명 * 6개의 동작 * 12번)이다.

3.2 제스처 EMG 데이터 설정 방안

제스처 분류를 위한 사용자 제스처 수집 시간 즉, 세그먼트 사이즈(Segment Size)는 1,000ms로 1 초안에 수집된 EMG 데이터로 제스처를 분류하는 것이 보다 명확하게 특징을 분류할 수 있다[18]. 하지만 사용자 행동이 3가지 단계에 따라 일어나는 연구[19]에 따르면 각 단계 지각(100ms), 인지(70ms), 모션(70ms)으로 이루어 지기 때문에 task에 대한 사용자가 피드백을 반응하는데 총 240ms의 시간이 걸리기 때문에 이에 대한 지각에 대한 행동은 200ms 이내의 데이터를 취합하여 이를 전처리하고 결과를 내기 위한 방안으로 기준이 될 수 있다. 또한, 과업별 입력 시간을 고려한 실험[20]에서 과업에 얼마나 시간이 들어갔는지를 측정하였는데 각 과업에 대한 성공률이 높기 위해서는 입력이 최소 192ms에서 최대 320ms이내의 범위에서 사용자 행동이 완성되기 때문에 약 200ms~300ms 사이에 사람은 반응을 할 수 없다. 따라서 하나의 입력을 300ms안에 형성할 수 있는 데이터로 잡아서 이를 전처리하고 판단한다면 실시간 입력에 대한 단위로 사용될 수 있다. 이를 위해 사용자의 제스처를 실시간으로 분류하기 위해서는 세그먼트 사이즈가 300ms보다 작은 200ms이내의 시간으로 해야 한다[21]. 따라서 본 논문에서는

실시간이 가능한 최대 세그먼트 크기인 200ms를 기준으로 이를 비교 평가한다.

Myo Armband로 수집한 EMG 데이터를 이용한 머신러닝 학습을 위한 세그먼트 사이즈를 다음과 같이 100ms(0.01초), 150ms(0.15초), 200ms(0.2초), 250ms(0.25초), 300ms(0.3초), 350ms(0.35초), 400ms(0.4초), 500ms(0.5초), 1000ms(1초) 총 9가지의 사이즈로 설정하여 짧아진 데이터 수집 시간에서도 제스처 분류가 가능한지를 연구한다. 또한, EMG 데이터 추출에서 disjoint segmentation과 overlapped segmentation 방법으로 데이터를 추출할 수 있다[22]. disjoint segmentation 방식은 데이터를 겹치지 않게 추출하는 방식이고, overlapped segmentation은 일정한 간격(increment)을 두고 데이터를 겹치게 추출하는 방식이다. 따라서 overlapped segmentation이 disjoint segmentation보다 더 많은 양의 데이터를 추출할 수 있다. 즉, 더 많은 양의 데이터를 추출하는 것은 더 다양한 형태의 데이터를 추출할 수 있기 때문에 학습하기 좋은 방안이다. 본 논문에서의 각 세그먼트 사이즈별 increment를 50ms로 설정하여 세그먼트 사이즈에 따라 표 1과 같이 샘플을 생성하였다. EMG를 수집한 시간은 동일하기 때문에 데이터 세그먼트 사이즈가 커질수록 총 샘플 개수는 줄어든다.

(표 1) 세그먼트 사이즈별 EMG 데이터 샘플 개수
(Table 1) Number of EMG data samples by segment size

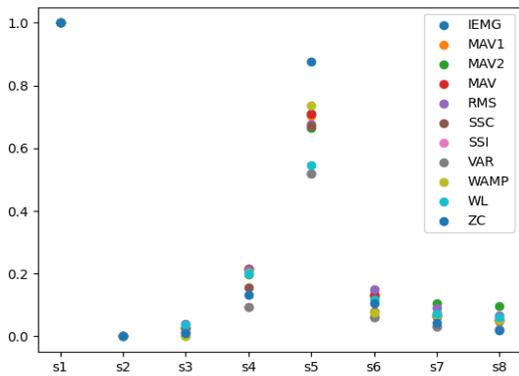
Segment size	Sample per a person(ea)	Total samples(ea)
100ms	2,808	14,040
150ms	2,736	13,680
200ms	2,664	13,320
250ms	2,592	12,960
300ms	2,520	12,600
350ms	2,448	12,240
400ms	2,376	11,880
500ms	2,232	11,160
1,000ms	1,512	7,560

4. 실험결과

4.1 전처리별 데이터 특징 비교

15가지의 EMG 전처리 과정에서 특징을 반영하지 못

하는 MAVS, AR, MDF, MNF는 제외한다[10]. 제외 후 IEMG, MAV1, MAV2, MAV, RMS, SSC, SSI, VAR, WAMP, WL, ZC를 전처리 후 센서별 데이터를 비교하였다. EMG 데이터 특징 추출을 위한 센서별 전처리 결과는 그림 3과 같다. 11가지의 전처리 방법은 전체적으로 비슷한 특징을 나타내고 있다. 특히, 센서4(s4), 센서5(s5) 그리고 센서6(s6)에서 전처리마다 차이가 있어, 머신러닝 학습에 따른 결과가 달라질 수 있다. 하지만 전반적으로 전처리된 데이터의 특징이 유사하기 때문에 학습 데이터로 사용이 가능하다.



(그림 3) 센서별 전처리 결과 비교

(Figure 3) Comparison of preprocessing results for each sensor

4.2 머신러닝 모델 별 세그먼트 사이즈에 따른 정확도 비교

11가지로 전처리된 전체 데이터 세트를 훈련(Train)과 테스트(Test)로 분리하기 위해 8:2의 비율(훈련 80%, 테스트 20%)로 훈련 세트와 테스트 세트를 분류한다. 훈련 세트를 5 - 폴드 교차 검증(5-fold cross validation)을 통해 과적합을 확인하고 테스트 세트를 학습한다. 위 과정을 10번 반복하여 전체 데이터 세트의 과적합을 확인하고 테스트 세트의 정확도 평균을 구한다. 또한 모델별 훈련은 그리드 서치(Grid Search)하여 찾은 모델별 베스트 파라미터(Best Parameter)로 학습하였으며 세그먼트 사이즈별 결과는 아래와 같다.

KNN의 경우 파라미터값을 $n_neighbors$ 값을 6, $weights$ 를 $distance^*$, p 는 1로 설정하였을 경우 가장 좋은 결과를 나타내었다. 표 2의 결과값을 보면 1,000ms의 경우 모든 전처리 방법에 대해서 약 99% 정도의 정확도를 나타내

었다. 따라서 세그먼트 사이즈를 줄어나감에 따라 정확도의 변화를 살펴볼 수 있다. 표 2에서 세그먼트 사이즈가 작아지면 모든 전처리 방법에서 정확도가 낮아짐을 알 수 있다. 특히, 제스처 입력에 대한 실시간 피드백을 위한 200ms이하에서 MAV2, SSC, WAMP를 적용했을 경우 최소 91.08%, 82.77%, 90.31%의 정확도로 다소 낮게 측정되었다. 반면에 IEMG, MAV1, MAV, RMS를 적용한 경우는 200ms ~ 100ms의 세그먼트에서 96% 이상의 높은 정확도를 보인다. 즉, 세그먼트 사이즈의 길이가 작아짐에 따라 학습될 수 있는 특징이 줄어들지만 실시간 사용자 입력을 세그먼트 사이즈인 200ms 이하에서도 좋은 결과를 나타낸다.

(표 2) KNN의 세그먼트 사이즈별 11가지 전처리 정확도 (%)

(Table 2) KNN's 11 preprocessing accuracy by segment size (%)

KNN	100 ms	150 ms	200 ms	250 ms	300 ms	350 ms	400 ms	500 ms	1000 ms
IEMG	96.71	98.07	98.92	99.37	99.63	99.76	99.88	99.95	99.99
MAV1	96.16	97.70	98.78	99.26	99.57	99.73	99.91	99.96	99.98
MAV2	91.08	93.33	95.50	96.47	97.20	97.68	98.32	98.88	98.98
MAV	96.70	98.06	98.92	99.36	99.63	99.76	99.88	99.95	99.99
RMS	96.57	98.03	99.03	99.38	99.72	99.78	99.94	99.94	99.98
SSC	82.77	89.09	93.01	95.44	97.11	98.11	98.81	99.45	99.47
SSI	94.13	96.22	97.68	98.34	99.15	99.38	99.60	99.72	99.70
VAR	94.13	96.22	97.68	98.34	99.15	99.38	99.60	99.72	99.70
WAMP	90.31	94.13	96.43	97.89	98.67	99.17	99.44	99.72	99.74
WL	94.51	96.54	98.15	98.85	99.39	99.66	99.84	99.88	99.93
ZC	92.56	95.42	97.37	98.37	99.00	99.18	99.50	99.77	99.75

SVC의 파라미터값은 kernel은 rbf, C값을 100, gamma를 scale일 때 가장 좋은 결과를 나타내어 이를 설정하였다. 표 3을 보면 KNN과 마찬가지로 1,000ms에서 가장 좋은 결과를 가지고 실시간 입력을 위한 200ms 이하의 경우에는 IEMG, MAV1, MAV, RMS가 98.77%, 98.68%, 98.77%, 98.61%로 가장 좋은 결과를 나타낸다. 하지만 KNN에 비해 전체적으로 정확도가 약 1%에서 2% 낮은 것을 볼 수 있다. 또한, KNN에서 세그먼트사이즈가 200ms일 때 WL이 98.15%인데 반하여 SVC인 경우에는 97.8%로 다소 떨어지고, 100ms일 경우에는 90.94%로 약 4%의 차이를 보인다.

(표 3) SVC의 세그먼트 사이즈별 11가지 전처리 정확도 (%)

(Table 3) SVC's 11 preprocessing accuracy by segment size (%)

SVC	100 ms	150 ms	200 ms	250 ms	300 ms	350 ms	400 ms	500 ms	1000 ms
IEMG	97.03	98.05	98.77	99.02	99.24	99.45	99.56	99.68	99.90
MAV1	96.73	97.98	98.68	98.81	99.17	99.38	99.45	99.61	99.84
MAV2	91.97	93.78	95.78	96.61	97.36	97.66	98.26	98.60	98.67
MAV	97.03	98.05	98.77	99.02	99.24	99.45	99.56	99.68	99.90
RMS	96.75	97.87	98.61	98.95	99.17	99.41	99.52	99.60	99.83
SSC	83.96	89.60	92.62	94.38	95.79	96.29	96.99	97.68	97.47
SSI	92.68	94.90	96.02	96.61	97.13	97.57	97.89	98.07	98.06
VAR	92.68	94.90	96.02	96.61	97.13	97.57	97.89	98.07	98.06
WAMP	90.94	94.05	95.94	96.57	97.44	97.58	97.87	98.39	98.43
WL	94.83	96.64	97.80	98.21	98.81	99.06	99.25	99.48	99.76
ZC	93.59	95.31	96.50	97.15	97.52	97.69	98.22	98.45	98.50

(표 4) RF의 세그먼트 사이즈별 11가지 전처리 정확도 (%)

(Table 4) RF's 11 preprocessing accuracy by segment size (%)

RF	100 ms	150 ms	200 ms	250 ms	300 ms	350 ms	400 ms	500 ms	1000 ms
IEMG	97.16	98.33	98.92	99.26	99.53	99.65	99.78	99.83	99.88
MAV1	96.78	97.97	98.88	99.23	99.40	99.63	99.76	99.81	99.91
MAV2	92.26	94.11	96.05	96.64	97.40	97.70	98.43	98.77	98.78
MAV	97.15	98.33	98.92	99.27	99.54	99.64	99.78	99.83	99.88
RMS	97.00	98.36	99.02	99.31	99.55	99.67	99.78	99.86	99.81
SSC	83.92	89.34	92.83	94.55	96.14	96.95	97.86	98.69	98.54
SSI	97.01	98.36	99.01	99.32	99.54	99.67	99.78	99.86	99.81
VAR	97.00	98.36	99.01	99.32	99.54	99.67	99.78	99.86	99.81
WAMP	90.50	94.04	95.98	97.15	98.04	98.60	98.91	99.35	99.29
WL	95.32	97.26	98.20	98.84	99.21	99.38	99.57	99.79	99.78
ZC	92.92	95.54	96.99	98.02	98.60	98.82	99.18	99.45	99.56

RF의 경우 $n_estimator$ 값을 100으로 지정할 때 가장 좋은 결과가 나타난다. 또한, XGBoost는 $learning_rate$ 값을 0.3, max_depth 의 값을 4, min_child_weight 값을 0으로 하였을 경우 가장 좋은 결과를 보여주었다. 표 4와 표 5의 결과를 볼 때 RF와 XGBoost는 IEMG, MAV1, MAV, RMS, SSI, VAR이 200ms 이하의 세그먼트 사이즈에서

최소 96%~최대 99.02%의 정확도 보여주는 것을 알 수 있다. 이를 종합해 볼 때 KNN, SVC, RF, XGBoost 모두 IEMG, MAV1, MAV, RMS를 사용할 경우 200ms 이하에서 98.72%이상의 높은 정확도를 가진다. 다만, XGBoost와 RF는 SSI, VAR에서도 98.75%의 높은 정확도를 보여 각 특징에 따른 머신러닝 방법을 알아내었다.

(표 5) XGBoost의 세그먼트 사이즈별 11가지 전처리 정확도 (%)

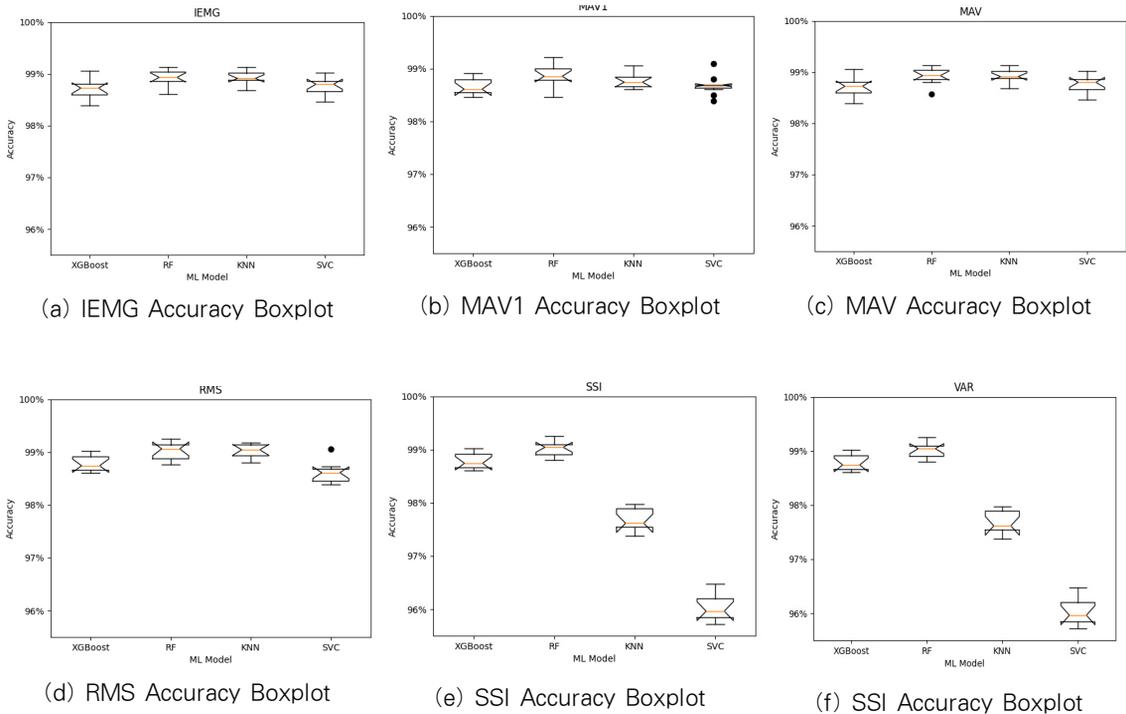
(Table 5) XGBoost's 11 preprocessing accuracy by segment size (%)

XG Boost	100 ms	150 ms	200 ms	250 ms	300 ms	350 ms	400 ms	500 ms	1000 ms
IEMG	96.86	97.98	98.72	99.19	99.46	99.62	99.74	99.86	99.81
MAV1	96.47	97.72	98.61	99.12	99.31	99.53	99.69	99.85	99.79
MAV2	91.80	93.62	95.71	96.32	97.10	97.59	98.19	98.52	98.49
MAV	96.85	97.98	98.72	99.19	99.46	99.62	99.74	99.86	99.81
RMS	96.68	97.98	98.75	99.18	99.45	99.60	99.75	99.84	99.70
SSC	83.64	88.83	91.98	93.70	95.58	96.41	97.34	98.06	98.08
SSI	96.68	97.99	98.75	99.18	99.45	99.60	99.75	99.84	99.70
VAR	96.68	97.99	98.75	99.18	99.45	99.60	99.75	99.84	99.70
WAMP	90.14	93.45	95.57	96.49	97.34	98.09	98.50	99.09	99.05
WL	94.77	96.67	97.70	98.38	99.04	99.18	99.50	99.68	99.71
ZC	92.71	95.08	96.61	97.52	98.39	98.49	98.96	99.28	99.33

4.3 전처리 방법에 따른 머신러닝 모델별 정확도 분포

앞서 제시한 실시간이 가능한 최대 세그먼트 크기인 200ms를 기준으로 정확성이 높은 6가지 (IEMG, MAV1, MAV, RMS, SSI, VAR) 전처리 방법에 대한 모델 별 안정성을 비교하기 위해 머신러닝 모델별 총 10번의 학습 후 정확도 분포를 박스 플롯(Box plot)으로 그림 4와 같이 나타낸다.

그림 4-a,b,c,d에서 IEMG, MAV1, MAV, RMS는 모든 머신러닝 모델에서 정확도가 98%에서 99% 사이에 전체적으로 밀집되어 있어 안정적인 결과를 보인다는 것을 알 수 있다. 다만, MAV1에서 SVC가 이상치의 값이 다수 발생하기 때문에 다른 모델들보다 다소 안정되지 않은 결과를 보여준다. 반면에 그림 4-e,f에서 SSI와 VAR의 경우 머신러닝 모델별 정확도가 KNN과 SVC의 경우 98% 이하의 정확도를 가지고 있지만 RF와 XGBoost인 경우에는 여전히 98%이상의 정확도를 나타내기 때문에 여전히



(그림 4) 모델별 안정성 비교 : (a)IEMG, (b)MAV1 (c)MAV, (d)RMS, (e)SSI, (f)VAR
 (Figure 4) Comparison of model stability : (a)IEMG, (b)MAV1 (c)MAV, (d)RMS, (e)SSI, (f)VAR

이 두 방법은 SSI와 VAR의 특징을 가지고도 사용될 수 있다. 이는 SSI와 VAR의 전처리 결과가 그림 3에서와 같이 전처리 결과가 거의 비슷하게 나오기 때문이다. 이를 통해 IEMG, MAV1, MAV, RMS와 같은 경우에는 4가지의 머신러닝 방법으로 모두 98%이상의 정확도를 나타낼 수 있고, SSI와 VAR은 RF와 XGBoost만 98%이상의 정확도를 나타낸다.

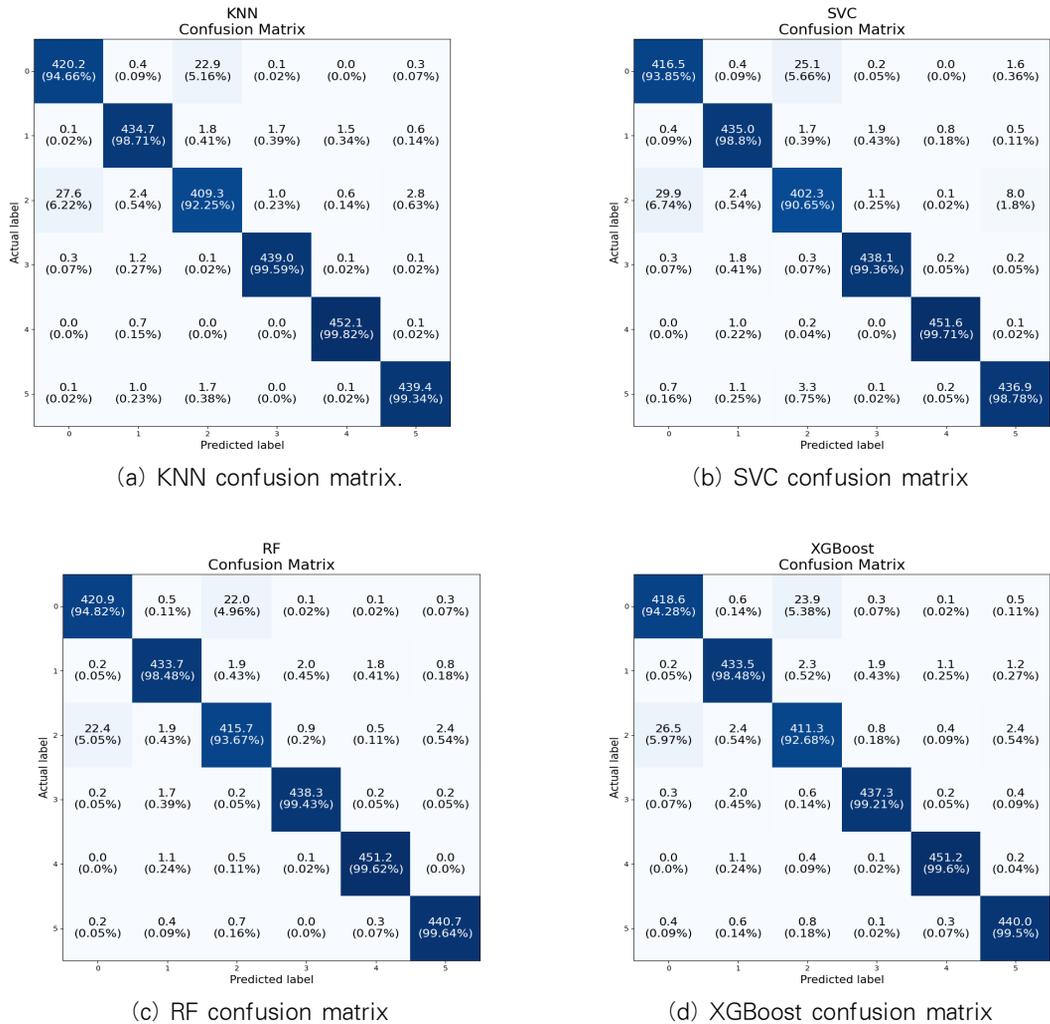
4.4 모델별 혼동행렬(confusion matrix) 비교

그림 5는 세그먼트 사이즈가 200ms일 때 머신러닝 모델별(KNN, SVC, RF, XGBoost) 혼동행렬을 나타낸다. 각각의 혼동행렬은 11개의 전처리의 제스처 예측 결과에 평균을 보여준다. 각 동작 유형별 정확도는 일정하게 높게 나타내고 있지만, 모든 모델과 모든 전처리 방법에서도 제스처 0번(wrist flexion)과 2번(hand at rest)을 구분할 때 정확도가 다소 떨어짐을 알 수 있다. 이는 그림 2에서 볼 수 있듯이 0번과 2번 동작의 유사성이 있기 때문이다.

하지만 동작을 오인하는 비율이 높지 않은 걸 확인할 수 있다. 0번 동작을 입력했을 때 2번 동작으로 판단하는 경우가 평균적으로 KNN 22.9회, SVC 25.1회, RF 22.0회, XGBoost 23.9회이고 2번 동작으로 입력했을 때 0번 동작으로 판단하는 경우가 평균 KNN 27.6회, SVC 29.9회, RF 22.4회, XGBoost 26.5회를 나타낸다. 이를 토대로 동작 0번을 입력했을 때 동작 2번이라고 오인하는 경우와 동작 2번을 입력했을 때 동작 0번으로 인식하는 경우가 평균적으로 모델에 따라 약 0.4~4.8회가 차이가 나는 것을 알 수 있다.

5. 결 론

EMG는 공간적 제약이 있거나 VR/AR과 같은 환경에서 사용하는 다양한 입력 인터페이스 중 직관적 인터페이스로 이용될 수 있다. 입력 인터페이스로 EMG를 사용하려면 사용자의 제스처를 실시간으로 입력 가능해야 하



(그림 5) 모델별 혼동행렬 비교 : (a)KNN, (b)SVC, (c)RF, (d)XGBoost

(Figure 5) Comparison of confusion matrices by model : (a)KNN, (b)SVC, (c)RF, (d)XGBoost

기 때문에 사용자로부터 EMG를 수집하는 시간을 어떻게 정해야 할지 판단할 필요가 있다. 따라서 본 논문에서는 사용자가 실시간 입력을 위한 시간에 따른 분류 정확도를 비교 분석하였다. 이를 위해 Myo Armband를 활용하여 사용자로부터 EMG 데이터를 수집하고, 이를 overlapped segmentation으로 9가지(100ms, 150ms, 200ms, 250ms, 300ms, 350ms, 400ms, 500ms, 1,000ms) 세그먼트 사이즈로 분할하고 11가지(IEMG, MAV1, MAV2, MAV, RMS, SSC, SSI, VAR, WAMP, WL, ZC) 전처리 방법을 통해서

특징을 추출하였다. 이렇게 생성된 특징을 4가지(KNN, SVC, RF, XGBoost) 머신러닝 방법을 통해 비교 분석하였다. 그 결과 공통적으로 IEMG, MAV1, MAV, RMS를 사용하고 KNN, SVC, RF, XGBoost를 사용할 때 정확도가 약 98% 이상인 것을 알아내었다. 또한, SSI와 VAR를 사용할 경우 XGBoost와 RF 모델을 사용할 경우에만 IEMG, MAV1, MAV, RMS와 비슷한 정확도를 가진다는 것을 알아내었다. 이러한 결과를 통해 사용자 제스처를 EMG를 통해 분류하여 입력하는 방안으로 사용하는 방안을 제시

하였다.

참고문헌(Reference)

- [1] Kaushik, Dr Manju, and Rashmi Jain “Gesture based interaction NUI: an overview,” International Journal of Engineering Trends and Technology (IJETT), Vol. 9, No.12, pp.633-636, 2014.
<https://doi.org/10.48550/arXiv.1404.2364>
- [2] Y. Song, D. Demirdjian, and R. Davis “Continuous body and hand gesture recognition for natural human computer interaction,” ACM Trans. Interactive Intell. Syst., Vol. 2, No.5, pp.1-28, 2012.
<https://doi.org/10.1007/s11277-022-10029-0>
- [3] Apple Vision Pro Document : online access 20231123:
<https://developer.apple.com/documentation/visionos/>
- [4] GD Morais, LC Neves, AA Masiero, and MCF de Castro “Application of Myo Armband System to Control a Robot Interface,” BIOSIGNALS, Vol. 4, pp.227-231, 2016.
<https://www.scitepress.org/papers/2016/57063/57063.pdf>
- [5] Phinyomark, Angkoon, and Erik Scheme. “EMG pattern recognition in the era of big data and deep learning,” Big Data and Cognitive Computing. Vol. 2, No. 3, , 2018.
<https://doi.org/10.1007/s11277-022-10029-0>
- [6] Gopal, P., Gesta, A., & Mohebbi, A. (2022). “A systematic study on electromyography-based hand gesture recognition for assistive robots using deep learning and machine learning models,” Sensors, 22(10), 3650.
<https://doi.org/10.3390/s22103650>
- [7] C. H. Lee, S. I. Kang, S. H. Bae, J. W. Kwon, and D. H. Lee “A Study of a Module of Wrist Direction Recognition using EMG Signals,” Journal of Rehabilitation Welfare Engineering & Assistive Technology, Vol. 7, No. 1, pp. 51-58 2013.
<https://koreascience.kr/article/JAKO201315262488693.page>
- [8] C. Savur and F. Sahin, “Real-Time American Sign Language Recognition System Using Surface EMG Signal,” 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), pp. 497-502, 2015.
<https://doi.org/10.1109/ICMLA.2015.212>
- [9] Spiewak, Christopher, et al. “A comprehensive study on EMG feature extraction and classifiers,” Open Access Journal of Biomedical Engineering and Biosciences. Vol 1. No. 1, pp. 1-10, 2018.
<http://doi.org/10.32474/OAJBEB.2018.01.000104>
- [10] Phinyomark, Angkoon, et al. “Evaluation of EMG feature extraction for hand movement recognition based on Euclidean distance and standard deviation,” In: ECTI-CON2010: The 2010 ECTI International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology. IEEE, pp.856-860, 2010.
<https://ieeexplore.ieee.org/document/5491586>
- [11] Abbaspour, Sara, et al. “Evaluation of surface EMG-based recognition algorithms for decoding hand movements,” Medical & biological engineering & computing, vol. 58, pp.83-100, 2020.
<https://doi.org/10.1007/s11517-019-02073-z>
- [12] Englehart, Kevin, and Bernard Hudgins. “A robust, real-time control scheme for multifunction myoelectric control,” IEEE transactions on biomedical engineering, Vol 50. No. 7, pp. 848-854, 2003.
<https://doi.org/10.1109/TBME.2003.813539>
- [13] Rhodes, Chris, Richard Allmendinger, and Ricardo Climent. “New interfaces for classifying performance gestures in music,” International Conference on Intelligent Data Engineering and Automated Learning. Cham: Springer International Publishing, pp. 31-42, 2019.
https://doi.org/10.1007/978-3-030-33617-2_4
- [14] Guo, G., et al. “Knn model-based approach in classification,” On The Move to Meaningful Internet Systems 2003, pp. 986 - 996, 2003.
https://doi.org/10.1007/978-3-540-39964-3_62
- [15] Jiang, Lai, and Runming Yao. “Modelling personal thermal sensations using C-Support Vector Classification (C-SVC) algorithm,” Building and Environment 99, pp. 98-106, 2016.
<https://doi.org/10.1016/j.buildenv.2016.01.022>

- [16] Zhou, Tao, et al. "Adapting random forest classifier based on single and multiple features for surface electromyography signal recognition," 2019 12th international congress on image and signal processing, biomedical engineering and informatics (CISP-BMEI). IEEE, pp.1-6, 2019.
<https://doi.org/10.1109/CISP-BMEI48845.2019.8965719>
- [17] Chen, Tianqi, and Carlos Guestrin. "XGBoost: A scalable tree boosting system," Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining. pp. 785-794, 2016.
<https://doi.org/10.1145/2939672.2939785>
- [18] Bagherian Azhiri, Reza, Mohammad Esmaili, and Mehrdad Nourani. "EMG-Based Feature Extraction and Classification for Prosthetic Hand Control," arXiv e-prints, 2021.
<https://doi.org/10.48550/arXiv.2107.00733>
- [19] Stuart Card, Thomas Moran and Allen Newell. "The Model Human Processor: An Engineering Model of Human Performance," Handbook of Perception and Human Performance, Vol. 2, No 45-1, 1986.
- [20] Christian Corsten, Bjoern Daehlmann, Simon Veelker, and Jan Brochers, "BackXPress: Using Back-of-Device Finger Pressure to Augment Touchscreen Input on Smartphones," In Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems, pp. 4654-4666, 2017.
<https://doi.org/10.1145/3025453.3025565>
- [21] CS, Nam and DR, Shin. "Force-touch measurement methodology based on user experience," International Journal of Distributed Sensor Networks, Vol. 14, No. 4, 2018.
<https://doi.org/10.1177/1550147718767>
- [22] Oskoei, Mohammadreza Asghari, and Huosheng Hu. "Support vector machine-based classification scheme for myoelectric control applied to upper limb," IEEE transactions on biomedical engineering, Vol. 55, No. 8 pp. 1956-1965, 2008.
<https://doi.org/10.1109/TBME.2008.919734>

● 저 자 소개 ●



박 하 제(Ha-je Park)

2023년~현재 인하대학교 소프트웨어융합공학과 학부생

관심분야 : 머신러닝, 딥러닝, 반도체 공정, etc

E-mail : tomorrow0422@naver.com



양 희 영(Hee-Young Yang)

2021년~현재 인하대학교 소프트웨어융합공학과 학부생

관심분야 : 머신러닝, 딥러닝, 반도체 공정, etc

E-mail : gmludud6128@naver.com

● 저 자 소 개 ●



최 소 진(So-Jin Choi)

2023년~현재 인하대학교 소프트웨어융합공학과 학부생
관심분야 : 머신러닝, 딥러닝, 반도체 공정, etc
E-mail : soso.jean@inha.edu



김 대 연(Dea-Yeon Kim)

2021년~현재 인하대학교 메카트로닉스공학과 학부생
관심분야 : 머신러닝, 딥러닝, 반도체 공정, etc
E-mail : kicter12@gmail.com



남 춘 성(Choon-Sung Nam)

2005년 상명대학교 소프트웨어학과(이학사)
2007년 숭실대학교 대학원 컴퓨터학과(공학석사)
2011년 성균관대학교 대학원 전자전기컴퓨터학과(공학박사)
2020년~현재 인하대학교 소프트웨어융합공학과 조교수
관심분야 : 무선 센서네트워크, 인터페이스 개발, 머신러닝, 딥러닝, 반도체 공정 etc.
E-mail : namgun99@inha.ac.kr